

第 4 课：运算符与表达式[上]

讲师：Monkey

时间：2016 年 5 月 6 日（周五）

课程要点：

1. “=” 号与 “+” 号
2. 算数运算符
3. “++” 与 “--”
4. 复合赋值运算符
5. 表达式

1. “=” 号与 “+” 号

1. “=” 号

表示赋值的意思，表示把等号右边的值，赋值给等号左边的变量。

演示：变量声明

2. “+” 号

在 C# 中 “+” 号有两个作用：

- ① 连接【当 “+” 号两边有一边是字符串的时候，“+” 号就起到连接的作用】
- ② 相加【当 “+” 号两边都是数字的时候，“+” 号就起到相加的作用】

演示：字符串+，数字+，混合+

2.算数运算符

1.+ - * /

加减乘除,四则运算。

优先级: 先乘除,后加减,有括号先算括号里的,相同级别的从左至右运算。

注意: 两个整数相除,结果会舍弃余数

演示: 计算三科总成绩和平均成绩

2.%

取模(取余数)

例: $10 \% 3 = 1$;

演示: 计算 365 是由几个 100,几个 10,几个 1 组成

3. “++” 与 “--”

1.前++ 后++

如果单独使用,不管是前++还是后++,最终的结果都是给这个变量加 1。

例: `number++ ++number`

如果是和表达式配合使用:

前++,先给这个变量自身加 1,然后带着这个加 1 后的值去参与运算。

后++,则先拿原值参与运算,运算完毕后,再将这个变量自身加 1。

例: `sum = ++number; sum = number++;`

2.前-- 后--

同上。

3.一元运算符与二元运算符

对于++和--这样只需要一个操作数就能完成的运算,我们称之为**一元运算符**。

+ - * / % 对于这些需要两个或两个以上操作数才能完成运算的操作符,我们称之为**二元运算符**。

4.复合赋值运算符

+= -= *= /= %=

例: `number += 10;` 效果等同于 `number = number + 10;`

5.表达式

概念: 由运算符相连的式子,就叫做表达式。